

Data Secrecy in Distributed Storage Systems under Exact Repair

Sreechakra Goparaju, Salim El Rouayheb, Robert Calderbank and H. Vincent Poor

Abstract—The problem of securing data against eavesdropping in distributed storage systems is studied. The focus is on systems that use linear codes and implement exact repair to recover from node failures. The maximum file size that can be stored securely is determined for systems in which all the available nodes help in repair (i.e., repair degree $d = n - 1$, where n is the total number of nodes) and for any number of compromised nodes. Similar results in the literature are restricted to the case of at most two compromised nodes. Moreover, new explicit upper bounds are given on the maximum secure file size for systems with $d < n - 1$. The key ingredients for the contribution of this paper are new results on subspace intersection for the data downloaded during repair. The new bounds imply the interesting fact that the maximum data that can be stored securely decreases exponentially with the number of compromised nodes.

I. INTRODUCTION

We study the problem of making distributed storage systems (DSS) information-theoretically secure against eavesdropping attacks. These systems are witnessing a rapid growth in recent years and include data centers and p2p cloud storage systems. These systems use data redundancy to achieve data reliability and availability in the face of frequent node failures. Three-times (3x) data replication has been the industry standard to achieve this goal. However, this solution does not scale well with the large amounts of data (in the order of petabytes) that these systems need to store. For this reason, data centers have started utilizing more sophisticated erasure codes on part of their data (typically the “cold” data that is not highly accessed) to protect against data loss [1], [2].

Erasure codes can achieve the same reliability levels as 3x replication with a much reduced storage overhead. However, they result in other system costs consisting of higher repair bandwidth, disk reads, computation complexity, etc. Moreover, erasure codes present new challenges when trying to secure the system. We illustrate this phenomenon with the example in Fig. 1, which depicts an $(n, k, d) = (4, 2, 2)$ DSS. The parameter $n = 4$ represents the total number of nodes of unit storage capacity each, and $k = 2$ is the number of nodes contacted by a user to retrieve the stored file. A new node, added to the system after a failure, contacts $d = 2$ other nodes to download its data (d is referred to as the repair

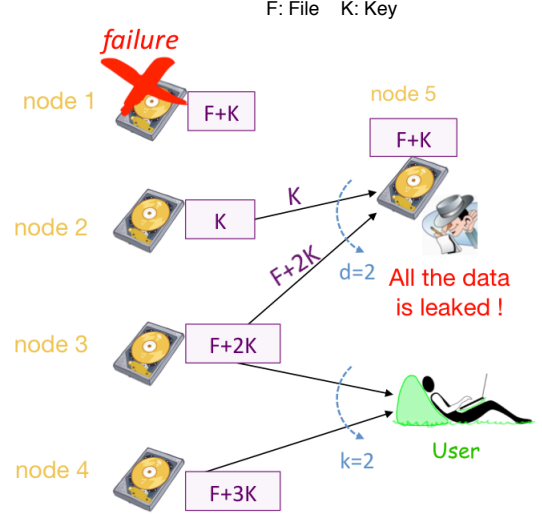


Fig. 1. An example of how repairing a DSS can compromise the system security. The original DSS formed of nodes 1, \dots , 4 is secured against a single compromised node using a secret sharing scheme or a coset code. However, repairing failed nodes can break the security of the system. For instance, consider the case when node 1 fails and is replaced by node 5, which is already compromised. The eavesdropper can observe all the data downloaded by node 5 and therefore decode the stored file F .

degree). Fig. 1 shows the failure and repair of node 1. Using a maximum distance separable (MDS) code, such as a Reed-Solomon code, the user can store a file of size 2 units in the DSS, which is also the information-theoretically optimal size. Now, suppose that we want to protect the system against an eavesdropper that can observe at most one node in the DSS unknown to us. If the system does not experience failures and repairs, then one can store securely a file F of one unit on the DSS by “mixing” the information file F with a randomly generated unit sized key K using the code depicted in the figure. This code can be regarded as a secret sharing scheme [3], a coset code for the wiretap channel II [4], or as a secure network code for the combination network [5], [6]. The code allows a user contacting any 2 node to decode the file F and leaks no information to the eavesdropper. A security violation occurs, however, when a node fails and is replaced by a new one. The replacement node has to download data from the surviving nodes in the system to regenerate the lost data. Now, if the new node is already compromised, this will reveal all the downloaded data to the eavesdropper. For instance, the figure depicts the case when node 1 fails and the coded data chunk $F + K$ is lost. The new replacement node downloads the two data chunks K and $F + 2K$ to decode the lost packet

S. Goparaju, S. El Rouayheb, and H. Vincent Poor are with the Department of Electrical Engineering, Princeton University, USA (e-mails: goparaju, salim, poor@princeton.edu).

R. Calderbank is with the Department of Computer Science, Duke University, USA (e-mail: robert.calderbank@duke.edu).

This research was supported by the U. S. National Science Foundation under Grant CCF-1016671.

$F + K$. However, this may reveal these two packets to the eavesdropper which can decode the file F . Therefore, even if we start with a perfectly secure code, the repair process can break the system security and result in data leakage. Our goal in this paper is to quantify how much data can be stored securely in a storage system even when the system experiences failures and repairs.

We consider systems that implement *exact repair*¹ in which the repair process regenerates an exact copy of the lost packet (see Fig. 1). Exact repair is a requirement in many practical systems for numerous reasons, such as preserving the systematic form of the data and allowing temporary reconstruction of data stored on a “hot” (i.e. highly accessed) node [1]. We also focus on *linear* coding schemes since they are the dominant class of codes employed in practice due to their ease of implementation. For these systems we are interested in quantifying the maximum amount of data that can be stored in a DSS with a given storage and repair bandwidth budgets while keeping the system *perfectly secure*. This means that we want to guarantee that no information is leaked to an eavesdropper that can observe a certain number of nodes in the system.

Contribution: We find an expression for the maximum file size that can be stored securely on a DSS under the linear coding and exact repair constraints. Our result holds for any number of compromised nodes for a DSS with repair degree $d = n - 1$. Similar results in the literature exist only for systems in which at most two nodes can be compromised by an eavesdropper. We also give new explicit upper bounds on the maximum secure file size for systems with $d < n - 1$. The key ingredients for our contribution are new results on subspace intersection for the data downloaded during repair. Our bounds imply the interesting fact that the maximum secure file size in the minimum storage regime decreases exponentially with the number of compromised nodes in contrast with for example the minimum-bandwidth regime [8], [9] or secret sharing schemes where it decreases linearly.

Related work: Dimakis et al. studied in [7] the information-theoretic tradeoff between storage overhead and repair bandwidth in distributed storage systems. Pawar et al. studied the problem of securing distributed storage systems under repair dynamics against eavesdroppers and malicious adversaries in [8], [10], [11]. They provided upper bounds on the system secure capacity and proved its achievability in the bandwidth-limited regime for repair degree $d = n - 1$. Shah et al. constructed secure codes based on the product-matrix framework in [9] and [12]. These codes can achieve the upper bound in [10] for the minimum-bandwidth regime and for any repair degree d . Rawat et al. gave tighter bounds on the secrecy capacity of a DSS in the minimum storage regime [13] and proved the achievability of their bound for $d = n - 1$ and for certain system parameters. Dikaliotis et al. studied the security of distributed storage systems in the presence of a trusted verifier [14].

¹ See [7] for the other type of repair referred to as *functional* in the literature.

Organization: The paper is organized as follows. In Section II, we describe the system and eavesdropper models and set up the notation. In Section III, we state our main results. We follow these by first providing an intuition behind the results in Section IV and then the proofs in Section V. We conclude with a summary of our results and open problems in Section VI.

II. PROBLEM SETTING

A. System Model

A distributed storage system consists of n *active* storage units or nodes $\{1, 2, \dots, n\}$, each with a storage capacity of α symbols belonging to some finite field \mathbb{F} . Nodes in a DSS are unreliable and fail frequently. When a storage node fails, it is replaced by a new node with the same storage capacity α . A DSS storing a data file \mathcal{F} of M symbols (in \mathbb{F}) allows any legitimate user called a data collector to retrieve the M symbols and reconstruct the original file \mathcal{F} by connecting to any k out of the n active nodes. We term this the *MDS property* of the DSS. Furthermore, we focus on single node failures since they are the most frequent in such systems. A new node added to the system to replace a failed one connects to d arbitrary nodes chosen out of the remaining $n - 1$ active ones and downloads β units from each. The repair degree d is a system parameter satisfying $k \leq d \leq n - 1$, and the nodes aiding in the repair are called *helper* nodes. The so-called *repair* process usually demands a higher repair bandwidth $d\beta$ than the amount of data α it actually stores. Moreover, the reconstructed data can possibly be different from the original data stored in the failed node. We define an (n, k, d) -DSS as a DSS that uses d nodes for the repair of a failed node to continuously maintain the k -out-of- n MDS property.

Dimakis et al. [7] showed that there is a fundamental tradeoff between the amount of data stored in each node α and the minimum repair bandwidth $d\beta$ required to store a file in the system. We focus on one extremity of this tradeoff, called minimum storage, in which each node stores the minimum possible $\alpha = M/k$. An MDS code achieving the minimum repair bandwidth for this α ,

$$d\beta = \frac{d\alpha}{d - k + 1}, \quad (1)$$

is referred to as an optimal bandwidth MDS code or a minimum storage regenerating (MSR) code. Furthermore, in this paper, we consider the case of *exact repair*, where the replacement node is required to reconstruct an exact copy of the lost data. In other words, the DSS consisting of n active nodes (and the MSR code) is invariant with time. It has been shown that optimal repair bandwidth is achievable for exact repair [15].

We concentrate on the practical scenario of linear MSR codes, which preserve the optimal repair bandwidth of (1). Without loss of generality, we can separate the nodes in the DSS storing an MDS code into systematic and parity nodes. We designate the first k nodes as systematic, where node $i, i \in [k] := \{1, 2, \dots, k\}$, stores the data vector w_i of

column-length α . The data vector w_{k+i} stored in parity node $i, i \in [n-k]$, is given by

$$w_{k+i} = \sum_{j=1}^k A_{i,j} w_j, \quad (2)$$

where $A_{i,j} \in \mathbb{F}^{\alpha \times \alpha}$ is the *coding matrix* corresponding to the parity node $i \in [n-k]$ and the systematic node $j \in [k]$. For optimal bandwidth repair of a failed systematic node $i \in [k]$, all other nodes transmit β amount of information, i.e., a helper node $j \neq i$ transmits a vector of length β given by $V_{j,i} w_j$, where $V_{j,i} \in \mathbb{F}^{\beta \times \alpha}$ is the repair matrix used for the repair of node i by node j . The vector $V_{j,i} w_j$ can also be interpreted as a projection of w_j onto a subspace of dimension β . We will use $V_{j,i}$, interchangeably, to denote both the matrix and the subspace obtained by the span of its rows.

B. Eavesdropper Model

We assume the presence of an eavesdropper Eve in the DSS, which can passively observe but not modify the contents of up to $\ell < k$ nodes of its choice. Eve can not only observe the data stored in a node i , but also the repair data $V_{j,i} w_j$ flowing into its replacement from a helper node $j \neq i$. In other words, not only does Eve have complete knowledge of w_i , it can potentially infer a part of w_j as well. In line with our assumption of repair of only systematic nodes, we assume that Eve can observe the repair data for only a subset of the systematic nodes², \mathcal{E}_d , where $\mathcal{E}_d \subseteq [k]$, and denote the rest of the observed nodes (for which it just observes the stored data) as \mathcal{E}_s , $\mathcal{E}_s \subseteq [n]$. The size of these subsets are denoted by $\ell_1 = |\mathcal{E}_s|$ and $\ell_2 = |\mathcal{E}_d|$, where $\ell_1 + \ell_2 = \ell$. Finally, we assume that Eve has complete knowledge of the storage and repair schemes implemented in the DSS.

C. Secrecy Capacity

Let U be a random vector uniformly distributed over $\mathbb{F}^{M^{(s)}}$, representing an incompressible data file with $H(U) = M^{(s)}$. Let W_i denote the random variable corresponding to the data w_i stored in node i , $i \in [n]$. Let us assume that a set \mathcal{D} of d helper nodes aid in the repair of node i . We denote the random variable corresponding to the data transmitted by a helper node $m \in \mathcal{D}$ for the repair of node i by $S_m^i(\mathcal{D})$, and the total repair data downloaded by node i by $S_{\mathcal{D}}^i$. We drop the \mathcal{D} in the notation and call these S_m^i and S^i , respectively, when the context is clear.

Thus, W_i represents the data that can be downloaded by a data collector when contacting node i and observable by Eve when $i \in \mathcal{E}_s$, while S^i represents the total data revealed to Eve when accessing a node $i \in \mathcal{E}_d$. Notice that the stored data W_i is a function of the downloaded data S^i . For convenience let us denote $\{W_i : i \in \mathcal{A}\}$ by $W_{\mathcal{A}}$, $\{S_j^i : j \in \mathcal{A}\}$ by $S_{\mathcal{A}}^i$, and $\{S^i : i \in \mathcal{A}\}$ by $S^{\mathcal{A}}$.

²Note that for securing the data, we do not store the original file on the systematic nodes, but rather the original file data encoded with random keys. However, we shall continue to refer to these nodes as systematic for convenience.

The MDS property of the DSS can be written as

$$H(U|W_{\mathcal{A}}) = 0, \quad (3)$$

for all $\mathcal{A} \subseteq [n]$, such that $|\mathcal{A}| = k$. To store a file U on the DSS perfectly secured from the eavesdropper Eve, we have the *perfect secrecy* condition,

$$H(U|W_{\mathcal{E}_s}, S^{\mathcal{E}_d}) = H(U), \quad (4)$$

for all $\mathcal{E}_s \subseteq [n]$, $\mathcal{E}_d \subseteq [k] \setminus \mathcal{E}_s$, and $|\mathcal{E}_s| + |\mathcal{E}_d| < k$.

Given an (n, k, d) -DSS with ℓ_1 and ℓ_2 compromised nodes (as described above) its linear coding secrecy capacity $C_s(\alpha)$, is then defined to be the maximum file size $H(U)$ that can be stored in the DSS using an optimal bandwidth MDS code for exact repair, such that the reconstruction property and the perfect secrecy condition simultaneously hold, i.e.,

$$C_s(\alpha) := \sup_{\substack{\mathcal{A}, \mathcal{E}_s, \mathcal{E}_d: \\ (3), (4) \text{ hold}}} H(U). \quad (5)$$

III. MAIN RESULTS

In this section, we state our main results. The proofs will follow in Section V after give a rough idea behind the results in Section IV. The following lemma provides a lower bound on the sum of subspaces³ associated with the repair bandwidth from a particular node, when it aids in the repair of multiple nodes.

Lemma 1: Consider an (n, k, d) -DSS in the systematic form with nodes having storage capacity α . Let nodes $[k]$ be the systematic nodes and let $V_{i,j}$ be the $\beta \times \alpha$ matrix associated with the exact repair of node j by node i . Then, for $d = n-1$ and for each $i \in [k]$, we have

$$\dim \left(\sum_{j \in \mathcal{A}} V_{i,j} \right) \geq \left(1 - \left(\frac{n-k-1}{n-k} \right)^{|\mathcal{A}|} \right) \alpha, \quad (6)$$

where $\mathcal{A} \subseteq [k] \setminus \{i\}$ and $V_{i,j}$ is the subspace corresponding to the matrix.

The next theorem gives an upper bound on the (linear coding) secrecy capacity $C_s(\alpha)$ for a given number of compromised nodes.

Theorem 2: Consider an (n, k, d) -DSS with a node storage capacity of α , which stores an optimal bandwidth linear MDS code for exact repair of systematic nodes. Suppose an eavesdropper gains access to the data stored in ℓ_1 nodes and the data downloaded during the repair of ℓ_2 systematic nodes, such that

$$\ell_1 + \ell_2 < k.$$

The achievable secure file size M^s for the given MSR code is then upper bounded by

$$M^s \leq (k - \ell_1 - \ell_2) \left(1 - \frac{1}{d - k + 1} \right)^{\ell_2} \alpha. \quad (7)$$

³The sum of subspaces B, C is defined as $B+C = \{b+c : b \in B, c \in C\}$.

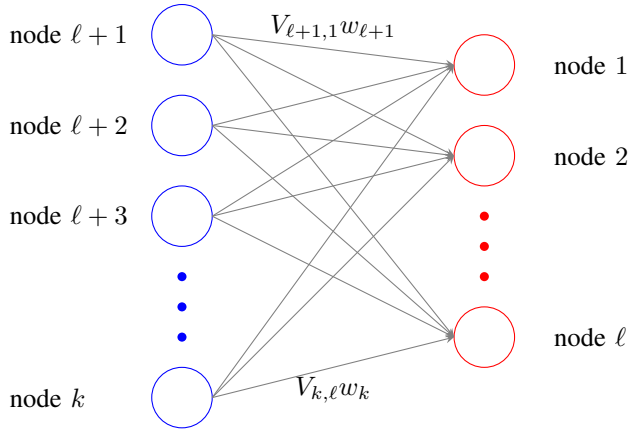


Fig. 2. An $(n, k, n-1)$ -DSS in which nodes $1, \dots, \ell$ have failed and have been replaced by the ℓ compromised nodes in red (nodes on the right). Each red node is repaired by contacting all of the other $d = n-1$ nodes in the system. For clarity, we only depict the edges between the red nodes and the remaining $k-\ell$ non-compromised systematic nodes (in blue, on the left). At a high level, the upper bound in Theorem 2 is obtained by evaluating the amount of information leaked to the eavesdropper in this scenario. This includes the information stored in the red nodes, plus the information contained in their downloaded data. The latter can be bounded using Lemma 1 which gives a handle on the correlation among all the data downloaded for repair.

The next theorem establishes that the upper bound in Theorem 2 is achievable when $d = n-1$.

Theorem 3: For an (n, k, d) -DSS with $d = n-1$, the secrecy capacity for optimal bandwidth MDS codes such that any systematic node is exact-repairable using a linear coding, is achievable for $\alpha = (n-k)^k$ and is given by

$$C_s(\alpha) = (k - \ell_1 - \ell_2) \left(1 - \frac{1}{n-k}\right)^{\ell_2} \alpha. \quad (8)$$

Moreover, the capacity is achievable for all (ℓ_1, ℓ_2) .

Proof: The proof follows from [13, Theorem 10] which describes an achievability scheme by precoding a (n, k) zigzag code [16] using a maximum rank distance code. ■

IV. SOME INTUITION

Before giving the formal proofs, we present in this section some intuition behind the upper bound (7) on the maximum achievable secure file size for a given DSS. We start with a simple toy example. How much data can we store securely in a system of 2 nodes of storage size α such that a user can recover it in the presence of an eavesdropper which has access to any one (unknown to us) node? We can quickly upper bound the answer by α data units by arguing that if we actually knew which node was compromised, we would not store any information in that node. In fact, this upper bound is achievable by using a random key r of size α and storing it on node 1 and storing $w + r$ on node 2, where w is the data. In other words, we subtract the amount of information visible to the eavesdropper from the total storage size available to the user. Then, by exploiting randomness this upper bound can be achieved even without knowing the identity of the compromised nodes.

We extend this argument to an (n, k, d) -DSS with nodes of storage capacity α and repair bandwidth β per helper node.

We explain our results for the specific case of two parity nodes and $d = n-1$ for which the optimal repair bandwidth $\beta = \alpha/2$. This means that each helper node sends half of its “information” to a replacement node. We also restrict our attention to the more-compromised nodes for which Eve can observe the repair data, i.e., let $\ell = \ell_2 = |\mathcal{E}_d|$, where $\mathcal{E}_d \subseteq [k]$. As in the 2-node example, we first try to find an upper bound on the maximum secure file size by asking how much would we store if we knew that the first ℓ nodes were compromised, or $\mathcal{E}_d = [\ell]$. We do not store any information on these ℓ nodes. We know however that Eve also gains some information about the nodes aiding in the repair of the compromised nodes if they were to fail. We assume that in a large enough length of time, each node fails at least once, and is repaired by the rest of the nodes. In particular, Eve has access to the information flowing to each of the compromised nodes from each of the remaining nodes. Fig. 2 shows the information flows which we shall focus on.

The information observable by Eve about node i , $i \in \{\ell+1, \dots, k\}$, is obtained by the vectors $V_{i,j}w_i$ communicated from node i to each compromised node $j \in [\ell]$. The total knowledge Eve has about node i is therefore equivalent to the combined information present in $\{V_{i,j}w_i\}_{j=1}^{\ell}$, or in other words, equivalent to the rank of the $\beta \times \ell\alpha$ matrix,

$$[V_{i,1} \mid V_{i,2} \mid \dots \mid V_{i,\ell}].$$

If instead, we view $V_{i,j}$ to be a subspace spanning the rows of the matrix $V_{i,j}$, this rank can also be represented as the dimension of the sum of subspaces,

$$\dim(V_{i,1} + V_{i,2} + \dots + V_{i,\ell}).$$

In this paper, we provide explicit bounds for the dimension of these sums of subspaces. For two parity nodes, we show that an addition of each repair subspace from a node reveals half of the information (about the helper node) which was unrevealed before the addition. To clarify, all subspaces reveal half of the information by design ($\beta = \alpha/2$). If we add two subspaces, because any two of these subspaces, say $V_{i,1}$ and $V_{i,2}$, cannot intersect in more than $\alpha/4$ dimensions [17], [13], their sum has to be more than

$$\frac{\alpha}{2} + \frac{\alpha}{4}$$

dimensions. Lemma 1 implies that for ℓ subspaces, a lower bound of

$$\frac{\alpha}{2} + \frac{\alpha}{4} + \dots + \frac{\alpha}{2^\ell} = \left(1 - \frac{1}{2^\ell}\right) \alpha$$

dimensions has to be revealed by node i in repairing the ℓ compromised nodes.

This calculation thus gives us the amount of information visible to Eve, which is $\ell\alpha$ from the compromised nodes and

$$(k - \ell) \left(1 - \frac{1}{2^\ell}\right) \alpha$$

from the $(k - \ell)$ non-compromised nodes. As in the 2-node example, it can be proved that using randomness (maximum

rank separable codes, [13]), we can securely store a total of $k\alpha$ minus the information visible to Eve, i.e.,

$$(k - \ell) \frac{1}{2^\ell} \alpha$$

data units in the presence of ℓ compromised nodes.

V. PROOFS

Proof of Lemma 1: We prove the lemma for the case of two parity nodes, i.e., $n = k + 2$. The proof can easily be extended to the case of more than two parity nodes. For the corresponding $(k, k + 2, d = k + 1)$ -DSS, as in Section II, we represent the symbols stored in the nodes $[n]$ by the column-vectors w_1, \dots, w_n of length α , and assume the first k nodes to be systematic. For convenience, we rename the coding matrices for parity node 1, $A_{1,j}, j \in [k]$ as $A_j, j \in [k]$, and those for parity node 2, $A_{2,j}, j \in [k]$ as $B_j, j \in [k]$.

When node j fails, node i transmits the matrix $V_{i,j}w_i$ in order to repair node j . When the number of parity nodes is 2, $V_{i,j}$ is an $\alpha/2 \times \alpha$ matrix. For notational simplicity, we represent the matrices $V_{k+1,j}$ and $V_{k+2,j}$ by $S_{1,j}$ and $S_{2,j}$ for all $j \in [k]$. It can be shown that an optimal bandwidth exact repair of systematic nodes necessitates interference alignment [18] and leads to the following *subspace conditions* (e.g. [17]):

$$S_{1,j}A_i \simeq S_{2,j}B_i, \quad (9)$$

$$\simeq V_{i,j}, \quad (10)$$

$$S_{1,j}A_j + S_{2,j}B_j \simeq \mathbb{F}^\alpha, \quad (11)$$

for all $j \in [k], i \in [k] \setminus \{j\}$, and \simeq denotes an equality of subspaces. In other words, the above subspace equalities specify the conditions required for the repair of a systematic node j by the set of helper nodes $[n] \setminus \{j\}$.

We prove the result stated in the lemma using induction. *Base case:* For $|\mathcal{A}| = 1$, we have $\dim(V_{i,j}) \geq \alpha/2$, which follows from the model constraints on the given DSS⁴.

Inductive step: Suppose the claim holds for $|\mathcal{A}| = m - 1$. We shall prove that the claim also holds for $|\mathcal{A}| = m$. Without loss of generality, let $\mathcal{A} = [m]$.

For $[k] \ni i \notin [m]$, we have

$$\dim \left(\sum_{j=1}^m V_{i,j} \right) = \dim \left(\sum_{j=1}^m S_{1,j}A_i \right), \quad (12)$$

$$= \dim \left(\sum_{j=1}^m S_{1,j} \right), \quad (13)$$

$$= \dim \left(\sum_{j=1}^m S_{1,j}A_m \right), \quad (14)$$

$$\geq \dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m \cap S_{2,m}B_m \right) + \dim(S_{1,m}A_m), \quad (15)$$

where (12) follows from (10), (13) and (14) follow from distributivity and the fact that the matrices A_i and A_m are invertible, and therefore $\dim(SA_i) = \dim(S) = \dim(SA_m)$, for any subspace S . For (15), notice that the subspaces

$$\left(\sum_{j=1}^{m-1} S_{1,j}A_m \right) \cap S_{2,m}B_m,$$

and $S_{1,m}A_m$ intersect only in the zero vector, see (11). Furthermore, both are contained in the subspace

$$\left(\sum_{j=1}^m S_{1,j}A_m \right),$$

and hence so is their *direct* sum.

Using the identity for arbitrary subspaces S_a and S_b , that $\dim(S_a + S_b) + \dim(S_a \cap S_b) = \dim(S_a) + \dim(S_b)$, and the fact that the subspaces $S_{2,m}B_m$ and $S_{1,m}A_m$ have dimension $\alpha/2$ (A_m and B_m being nonsingular), we obtain from (15),

$$\dim \left(\sum_{j=1}^m V_{i,j} \right) \geq \dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m \right) + \alpha - \dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m + S_{2,m}B_m \right). \quad (16)$$

The third term on the right hand side in inequality (16) equals the term on the left hand side, because

$$\begin{aligned} & \dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m + S_{2,m}B_m \right) \\ &= \dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m B_m^{-1} + S_{2,m} \right) \\ &= \dim \left(\sum_{j=1}^{m-1} S_{2,j} + S_{2,m} \right) \\ &= \dim \left(\sum_{j=1}^m V_{i,j} \right), \end{aligned} \quad (17)$$

where the steps follow from similar reasons as in (12)–(15). Also, similarly,

$$\dim \left(\sum_{j=1}^{m-1} S_{1,j}A_m \right) = \dim \left(\sum_{j=1}^{m-1} V_{m,j} \right). \quad (18)$$

Using the induction hypothesis and (16)–(18), we have

$$\begin{aligned} 2 \dim \left(\sum_{j=1}^m V_{i,j} \right) &\geq \dim \left(\sum_{j=1}^{m-1} V_{m,j} \right) + \alpha \\ &\geq \left(1 - \frac{1}{2^{m-1}} \right) \alpha + \alpha, \end{aligned}$$

which completes the inductive step. ■

⁴It can be shown from the MDS property of the storage code that all the coding matrices $\{A_i, B_i\}, i \in [k]$ have full rank, and from the subspace conditions that all the subspaces $V_{i,j}$ have full rank as well.

For the sake of completeness, we present here the information-theoretic proof given in [13] which transitions into the proof of Theorem 2 via Lemma 1. However, our notation, described in Section II, is inspired by [19].

Proof of Theorem 2: Let \mathcal{R} be any set of $k - \ell_1 - \ell_2$ systematic nodes not in \mathcal{E}_s or \mathcal{E}_d . In order to store a file U of entropy $M^{(s)}$ securely in the DSS, we have

$$M^{(s)} = H(U | W_{\mathcal{E}_s}, S^{\mathcal{E}_d}), \quad (19)$$

$$= H(U | W_{\mathcal{E}_s}, S^{\mathcal{E}_d}) - H(U | W_{\mathcal{E}_s}, S^{\mathcal{E}_d}, W_{\mathcal{R}}), \quad (20)$$

$$= I(U; W_{\mathcal{R}} | W_{\mathcal{E}_s}, S^{\mathcal{E}_d}), \quad (21)$$

$$\leq H(W_{\mathcal{R}} | S^{\mathcal{E}_d}), \quad (22)$$

$$\leq \sum_{i \in \mathcal{R}} H(W_i | S_i^{\mathcal{E}_d}), \quad (23)$$

$$= \sum_{i \in \mathcal{R}} (H(W_i, S_i^{\mathcal{E}_d}) - H(S_i^{\mathcal{E}_d})), \quad (24)$$

$$= \sum_{i \in \mathcal{R}} (H(W_i) - H(S_i^{\mathcal{E}_d})), \quad (25)$$

where (19) is the same as (4), (20) follows from (3) and the fact that W^i is a function of S^i , and (25) from the fact that S_i^m is a function of W_i , for any $m \neq i$. Using the linearity of the MDS code being used, we have

$$H(S_i^{\mathcal{E}_d}) = \dim \left(\sum_{j \in \mathcal{E}_d} V_{i,j} \right), \quad (26)$$

where because $i \in \mathcal{R}$, we have $\mathcal{E}_2 \subseteq [k] \setminus \{i\}$. Thus, we have,

$$M^{(s)} \leq (k - \ell_1 - \ell_2) \left(1 - \frac{1}{n - k} \right)^{\ell_2} \alpha. \quad (27)$$

For $d < n - 1$, we focus on the first $d + 1$ nodes, viewing it as an $(n' = d + 1, k, d = n' - 1)$ -DSS. The conditions of exact repair for this restricted system form a relaxation of the original problem, and thus an upper bound on $M^{(s)}$ for this system also holds for the latter. By the optimal bandwidth condition (1), and because exact repair requires interference alignment, from Lemma 1, we obtain for $i \in [k]$,

$$\dim \left(\sum_{j \in \mathcal{A}} V_{i,j} \right) \geq \left(1 - \left(\frac{d - k}{d - k + 1} \right)^{|\mathcal{A}|} \right) \alpha,$$

for $\mathcal{A} \subseteq [k] \setminus \{i\}$. Note that our helper set of nodes is $\mathcal{D} = [n'] \setminus \{i\}$ for repairing node i . We therefore obtain the required bound on $M^{(s)}$ using a similar set of equations as for $d = n - 1$. ■

VI. CONCLUSION

We have studied the problem of securing data in distributed storage systems against eavesdropping. Our focus has been on systems that implement linear codes and exact repair. We have determined the maximum file size that can be stored securely in these systems for any number of compromised nodes, when the repair degree $d = n - 1$. For the other cases, i.e., when $d < n - 1$, we have given new upper bounds on

the amount of secure data that can be stored in the system. Many questions remain open, such as constructing codes that can achieve our upper bound (7) for $d < n - 1$, and finding a general expression of the system secrecy capacity without the linearity and exactness assumptions.

REFERENCES

- [1] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proc. 2012 USENIX Annual Technical Conference (ATC)*, (Boston, MA), 2012.
- [2] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data," in *arXiv:1301.3791*, 2013.
- [3] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [4] L. H. Ozarow and A. D. Wyner, "Wire-tap channel-II," *AT&T Bell Lab Tech. Journal*, vol. 63, no. 10, 1984.
- [5] N. Cai and R. W. Yeung, "Secure network coding on a wiretap secure network coding on a wiretap network," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 424–435, 2011.
- [6] S. El Rouayheb and E. Soljanin, "Secure network coding for wiretap networks of type ii," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1361–1371, 2012.
- [7] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sep. 2010.
- [8] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 58, pp. 6734–6753, March 2012.
- [9] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," in *Proc. IEEE Global Communications Conference*, (Houston, TX), December 2011.
- [10] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," in *Proc. IEEE International Symposium on Information Theory*, (Austin, TX), 2010.
- [11] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems from malicious nodes," in *Proc. IEEE International Symposium on Information Theory*, (St. Petersburg, Russia), 2011.
- [12] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Regenerating Codes for Errors and Erasures in Distributed Storage," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, (Cambridge, MA), 2012.
- [13] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," in *arXiv:1210.6954*, 2013.
- [14] T. K. Dikaliotis, A. G. Dimakis, and T. Ho, "Security in distributed storage systems by communicating a logarithmic number of bits," in *Proc. IEEE Internat. Symp. Inform. Th. (ISIT'10)*, (Austin, TX), 2010.
- [15] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *arXiv:1004.4438*, 2010.
- [16] I. Tamo, Z. Wang, and J. Bruck, "Zigzag Codes: MDS Array Codes With Optimal Rebuilding," *Information Theory, IEEE Transactions on*, vol. 59, pp. 1597–1616, march 2013.
- [17] I. Tamo, Z. Wang, and J. Bruck, "Access vs. bandwidth in codes for storage," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 1187–1191, july 2012.
- [18] N. Shah, K. Rashmi, P. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *Information Theory, IEEE Transactions on*, vol. 58, pp. 2134–2158, april 2012.
- [19] N. Shah, K. Rashmi, P. Vijay Kumar, and K. Ramchandran, "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," *IEEE Transactions on Information Theory*, vol. 58, pp. 1837–1852, March 2012.